Proceedings of the 36th Israel Annual Conference on Aerospace Sciences. February, 21-22, 1996.
Haifa, Israel.

MODES IN AUTOMATED COCKPITS: PROBLEMS, DATA ANALYSIS, AND A MODELING FRAMEWORK

Asaf Degani
San Jose State University
and NASA Ames Research Center
Moffett Field, CA

Michael Shafto
and NASA Ames Research Center
Moffett Field, CA

Alex Kirlk
Georgia Institute of Technology
Atlanta, GA

ABSTRACT

Today's modern automated flight control systems employ a variety of different behaviors, or modes, in order to allow the crew flexibility in operating the aircraft. These modes lay along a continuum starting from fully-manual to fully-automatic control, with many levels in between. While developments in cockpit automation resulted in workload reduction and economical advantages, they also gave rise to a special class of human-machine problems. These ill defined problems are sometimes referred to as "automation surprises." These phenomena usually involve confusion about the status of the autoflight control system—in particular its modes—and the subsequent behavior of the aircraft. Recently, there have been five fatal airline accidents involving, in one way or another, this class of human-machine problems.

Our research involves a two-way approach: (1) summary analysis of mode usage data, and (2) the development of a methodology for describing human-automation interaction. The mode usage data was collected during a field experiment that documented how pilots, flying a Boeing B-757/767, engaged the various modes in the automated flight control system. Analysis of the data indicated which modes are commonly used, and the different paths that pilots use while transitioning from one mode to another. Based on these and other analyses we developed a modeling framework for describing human interaction with automated control systems. The framework allows us to formally describe a given system and to trace down the features in the control system or interface design that harbor the potential for mode confusion. One example of mode confusion is described and modeled. The particular design feature that led to this mode confusion is identified and explained. The implication of this framework and our research approach for mapping human-automation interaction are briefly discussed.

INTRODUCTION

In recent years there has been a series of commercial aviation accidents involving highly automated "glass cockpit" aircraft. In all of these accidents, the aircraft software and hardware functioned properly, yet the system as a whole, failed. The common cause of accident was some confusion, on part of the flight crew, about the status of the autoflight control system.

Since 1988, five such accidents occurred: In the first, an Air France Airbus A-320 crashed in Habersheim-Mullhouse Airport, France, following a low altitude fly-by [4]. The crew, flying very close to the ground, engaged a mode that provides relatively slow thrust response to throttle movement. In the second accident, an Indian Airlines A-320 crashed during a visual approach to Bangalore Airport, India [11]. The crew, intentionally or unintentionally, engaged a pitch mode in a way that provided no speed protection. In the third accident, an Air Inter A-320 crashed during an approach into Strasbourg-Entzheim Airport, France. The preliminary accident report suggests that the crew may have mistakenly engaged the wrong mode for the situation at hand [5].

In the fourth accident, a China Airlines A-300/600 crashed during an approach into Nagoya International Airport, Japan. The crew, probably unintentionally, engaged a mode that commanded climb with full thrust, and at the same time *manually* pushed the control wheel down in order to prevent the aircraft from climbing. In a conflict between manual versus autopilot commands, the aircraft achieved an extreme pitch attitude of 36 degrees with decaying airspeed, rolled to the right, and crashed [2]. In the fifth accident, an Airbus A-330 conducting a routine flight test crashed shortly after takeoff from Toulouse-Blagnac airport, France. During the initial climb-out, and according to the test-flight program, the crew reduced power on the left engine to simulate an engine-out go-around. With the reduction in thrust on the left engine, and a high pitch attitude (32 degrees) to acquire the 2,000 feet selected by the crew, the aircraft lost lateral control. The high pitch altitude occurred because the altitude acquisition mode, which was controlling the level-off maneuver to 2,000 feet, had no pitch limit protection. When the crew regained control of the aircraft, it was at too low an altitude to allow full recovery [6].

The way pilots use modes and the implications for managing the flight are issues of much discussion and concern in the commercial aviation community [3]. Indeed, airlines address this issue by designing specific training modules, developing specific procedures, and articulating a philosophy for using flight-deck automation [9]. Nevertheless, evidence from accidents and thousands of pilot-reported incidents suggests that confusion regarding mode behavior is still a chronic problem in these human-machine systems [10]. It also appears that this trend is bound to escalate as newer automated systems are developed and put into use [29].

Modes in Human-Machine Systems

Problems in human interaction with automated control systems are not unique to aviation; similar problems are found in medical, maritime, nuclear, and space systems [18]. In the domain of human-computer interaction, mode confusion is also a well-recognized

problem [24,30]. Nevertheless, even the common VCR is still not free of mode problems—most users are still baffled when they need to "program" a VCR.

What are the characteristics, then, of these systems that make them prone to such problems? The following is a partial list:

1. Such systems are designed to perform many functions and not just one or a few.
2. The functions are either activated directly by the user or by some external event(s).
3. For simplicity and "cleanness," the interface includes fewer control and display elements (e.g., buttons, icons) than the number of functions and interactions possible.
4. In the case of dynamic systems, the user must specify ("program") in advance what type of behavior the system should exhibit in the future.
5. The user must be informed when these specifications cannot be fulfilled due to some external or internal change.
6. The user must be able to intervene and modify the specifications at any time.

Most software system designed today use modes—either explicitly or implicitly—in order to achieve the above functionality [21]. We broadly define a mode as a *manner of behaving*. This general definition satisfies the use of the term within any system, be it a biological, social, or software and hardware system. Such systems behave in a certain manner, as a function (in the mathematical sense) of the environment [23]. Events in the environment initiate *modes* within the system—the transformations from one manner of behaving to another [1].

In complex systems the definitions become convoluted as a given machine, or a control system in our case, is usually made of several sub-components, working in parallel. Each of these components may have its own set of modes. Therefore, unlike a simple system that may exhibit one mode at a time, the status of a complex system, with respect to its modes, is a vector of all active modes. We use the term *mode configuration* to describe this vector. The automatic flight control system (AFCS) of a Boeing B-757 aircraft, exhibits numerous mode configurations. Transitions from one configuration to another can be either initiated *manually* by the pilot or *automatically* by the machine. Similarly, the system software initiates automatic transitions among various mode configurations depending on pre-defined conditions.

An indivisible attribute of any mode in dynamic systems is its reference-parameter(s) [16]. For example, "speed" is a reference parameter of most vertical guidance modes of a modern aircraft. Reference parameters are constraints on mode behavior and can be either continuous or a set of discrete states. This observation leads to an important insight: in almost any system: there are several classes, or a hierarchy, of modes. For example, the modern AFCS has two cardinal manners of behavior: "On" and "Off"—this is the first level. When in the "On" mode, the behavior is constrained by another set of modes: "Flight Level Change" mode, "Vertical Navigation" mode, and others—this is the second

level. When in "Flight Level Change" mode, one constraint on behavior is the speed setting (a reference parameter) provided by the pilot. When in "Vertical Navigation" mode, the behavior is further constrained by a set of sub-modes: "Vertical Navigation-Path," "Vertical Navigation-Speed" and others—this is the third level of modes in this system.

Problems in human-automation interaction, and in particular mode confusion, usually result from misidentification of the machine's behavior—its mode configurations, transitions, and the associated reference parameters. Such mode confusion results from errors—the difference between actual machine behavior and what was expected by the pilot. Some of these mode errors, or discrepancies, occur when the user takes some action (e.g., changing a reference parameter) believing that the machine is in one mode, when in fact it is in another [25].

Objectives

Transition between one mode configuration and another is a critical ingredient for mode confusion [17]. Therefore, the first objective of this paper is to analyze mode usage data in a context of the automatic flight control system of a Boeing B-757/767 aircraft. The second objective is to present a framework for modeling human-automation interaction. The third objective is to use the framework to describe and model a mode confusion incident and identify the specific design feature that contributed to it.

MODE USAGE DATA

Method

Data was collected by an observer onboard an airliner during the climb-to-cruise and descent-to-land phases of a flight. The data presented here is based on the analysis of 30 flights. Subjects were airline pilots from a major US carrier, flying regular revenue flights in either the Boeing B-757 or B-767 aircraft—both modern "glass cockpit" aircraft equipped with an identical automatic flight control system (AFCS). The AFCS is composed of three major components: autopilot, autothrottle, and flight management computer. Sitting in the cockpit jumpseat, the observer recorded a variety of AFCS and aircraft variables. The following variables were used: changes in pitch and roll modes, thrust modes, and whether the autopilot, flight-director, and autothrottle were "On" or "Off."

Analysis

Two types of data summaries were performed: mode occupancy and mode transitions. In the former we wanted to see which mode configurations are commonly engaged; in the latter, we were interested in identifying the transitions between these mode configurations. (See [19] and [22] for a similar approach for describing human interaction with complex systems.)

Mode occupancy. This analysis describes the relative frequency of occupying a certain mode configuration. The AFCS has a variety of roll and pitch modes, and the pilot may select any configuration. Figure 1 presents a table of these pitch and roll modes: the

columns list the five modes of the roll component, and the rows list the eight modes of the pitch component. Each cell in the table indicates a unique combination.

As mentioned earlier, mode configurations also indicate the level of automation selected by the pilot. At one extreme, the crew may decide to fly the aircraft completely manually (autopilot and autothrottle are disengaged, and the pilot is flying without reference to the flight director guidance). In this case the “Manual Roll” mode and “Manual Pitch” modes are engaged (upper-left corner of the table). At the other extreme, the crew may decide to fly the aircraft completely automatically (autopilot and autothrottle are engaged, and the aircraft follows the guidance from the flight management computer). In this case the “Lateral Navigation” mode and “Vertical Navigation” (lower-right corner) are engaged. Between these two extremes, 38 other mode configurations exist. The mode usage data was entered and coded into a dataset. The number of times each mode combination was visited during all flights was summarized and then transformed into relative frequencies.

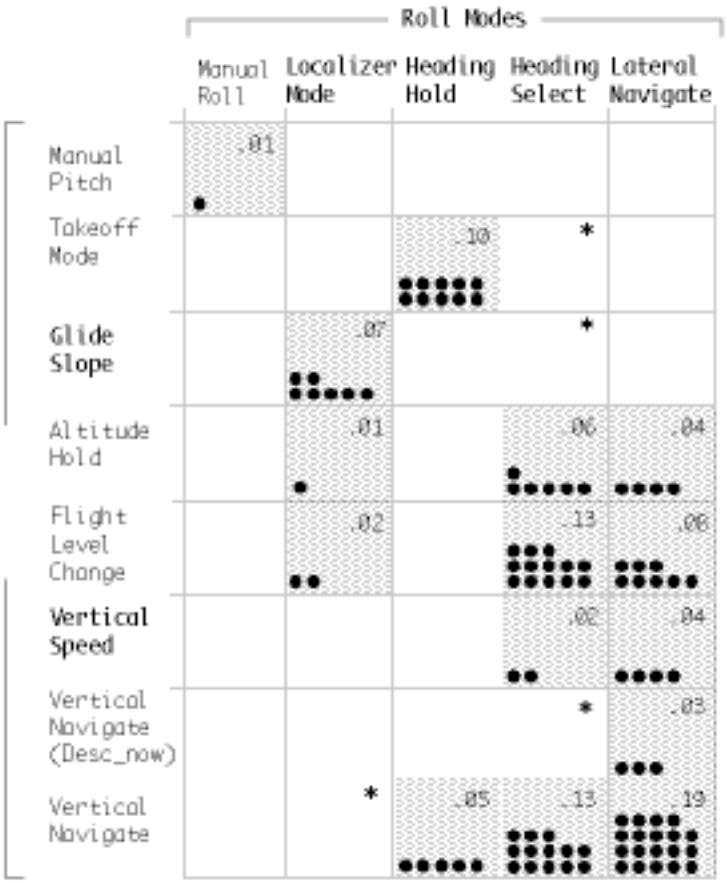

Figure 1. Mode occupancy (* indicate 0 < occupancy < 0.01)

The table presented in Figure 1 is a map of the AFCS mode configuration space; superimposed on it are the relative frequencies of occupying a given mode configuration. Two observations can be made from this table: (1) only about 50 percent of the pitch/roll mode configurations are occupied (at the 0.01 level), and (2) heavy occupancy is found in three mode configurations: "Takeoff Mode"/"Heading Hold," "Flight Level Change"/"Heading Select," and "Vertical Navigation"/"Lateral Navigation." The first mode configuration is associated with a mandatory procedure in the Airline—takeoffs are only performed by engaging this mode configuration. The second is a semi-automatic mode configuration in which the pilots can respond quickly to the type of air-traffic-control demands prevalent in the terminal area; it is also a recommended mode configuration (per standard operating procedures) when the aircraft is below 10,000 feet. The third mode configuration is fully automatic mode combination that provides fuel efficiency and navigation accuracy.

Mode transitions. This analysis describes the frequency of transitioning from one mode configuration to another. Transitions may be initiated by the control system (automatic transitions) or by the pilots (manual transitions). In both cases, the mode configuration space and associated transitions are constrained by external factors. For example, the transition to the "Glide Slope" mode can only occur after the glide slope signal is received and processed by the AFCS. Therefore, during the course of flight the mode configuration space changes dynamically.

The number and direction of transitions among all mode configurations were summarized and then transformed into relative frequencies. Based on the previous analysis, we only regarded those mode configurations that were occupied at the 0.01 level. These 15 mode configurations, depicted as nodes, and transitions among them are presented in Figure 2.

The transition diagram in Figure 2 combines two qualities: mode configuration and mode transition. It shows the possible paths that pilots traverse through the AFCS mode configuration space. One objective of this analysis was to identify the constraints on the flow of transitions. We hypothesized that this flow will reveal the pattern behind the interaction between the pilots and the AFCS. We have tried several schemes for organizing the nodes and the transitions. Yet, they all produced a criss-cross ("spaghetti") of arcs. Finally, we decided to let the data speak for itself [31]: we arranged the nodes according to one simple rule—minimize the number arc intersections. When this was finally done, a unique and unexpected pattern emanated.

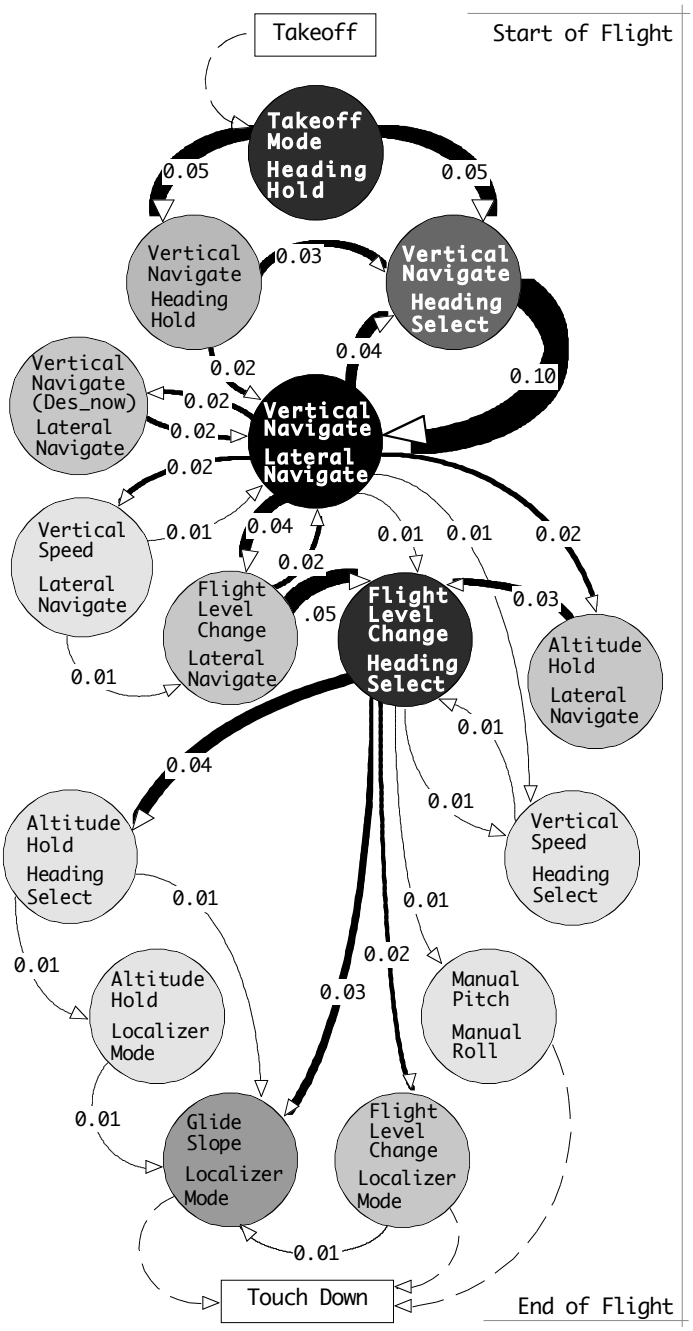

Figure 2. Mode transitions

The organization of nodes that best satisfied the rule was according to the phases of a flight. Thick arcs indicate many transitions into the “Lateral Navigation”/“Vertical Navigation” mode configuration. This is a fully automatic mode configuration that crews try to get into as fast as possible, as it provides many economic advantages. It serves as a pivot, or an attractor, in this diagram. Once in the descent phase, the “Flight Level Change”/“Heading Select” mode configuration is commonly used. In the approach phase, a gradual transition into the “Glide Slope”/”Localizer” mode configuration is observed.

Nevertheless, other transitions into touch-down were also observed. One such transition, however—via the "Flight Level Change"/"Localizer" mode configuration—is potentially dangerous at low altitudes.

The two summary analyses described in this section shed light on how modes are used during normal revenue flights. We identified the mode configurations commonly used by flight crews and the transitions among them. The data also revealed the flow of transitions from takeoff to touchdown. Taken as a whole, the summary analysis showed the pattern of pilot interaction with the automated flight control system of the Boeing B-757/767 aircraft.

A FRAMEWORK FOR MODELING HUMAN-AUTOMATION INTERACTION

From the above and other analysis performed on the field data, it appeared that mode transitions are constrained by several factors: the airline's standard operating procedures, the phase of flight, the demands from the environment (e.g., air traffic control clearances) and the possible mode configurations of the system. Therefore in order to model human interaction with modes, we needed to synthesize these factors into a single framework. The language of the framework is described first, then the elements, or modules, of the framework are presented.

Language

The framework is based on the Statecharts and Operator-Function models. Both share the same underlying theory—the finite-state-machine—which provides a medium for describing the control behavior of a given system in terms of its states and transitions [20]. The Operator Function Model is a task decomposition formalism used for task-analysis and simulation [14]. Depending upon demands from the operational environment and the requirements of the task (e.g., mandatory procedures), the model specifies the progression of functions, tasks, and actions that a well-trained operator will execute. Statecharts is a specification language for complex, real-time, reactive systems [12, 13]. Statecharts formalism allows for hierarchy and concurrency. Hierarchy is represented with the notion of a super-state and sub-states; concurrency is provided by the notion of independence and orthogonality such that a super-state containing two orthogonal sub-states can be described as their product (.AND. relationship). Another feature of Statecharts is its broadcasting mechanism—broadcasting the active states and events (that trigger transitions) to the entire network.

These features of the Operator Function Model and Statecharts are well suited for describing human interaction with a modal system: a finite-state-machine approach maps quite well for describing modes and transitions; hierarchy allows us to deal with the levels of modes in the machine; concurrency allows us to describe a multi-component system in which several modes are simultaneously active; and broadcasting allows us to connect, via transitions, all components into a synchronized system. Both the Operator Function Model and Statecharts represent the control behavior of the system—what is responsible for making the time-dependent decisions that influence human and machine behavior.

Modules in the Framework

This framework, called "Ofan," uses the Statecharts language to describe the human-machine-environment system as five concurrently active modules [7]:

(1) the *Environment*

(2) the *Human Functions/Tasks*,

(3) the *Control*s,

(4) the *Plant*

(5) the *Displays*

The *Environment* module describes the events in the operational environment as a collection of concurrent states. Although we realize that this view of the environment is oversimplified, it serves well for feeding triggering events into other modules in the framework. The *Human Functions/Tasks* module describes how the operator executes functions, tasks, and actions (modeled as states) and the events that trigger them. The *Controls* module describes the system's interface with the operator (e.g., buttons, dial knobs). The *Plant* module describes the control system—its components, states, inputs, and transitions. Closing the loop is the *Displays* module which describes the display features as perceived by the operator.

There are many ways one can model a system. The selection of the modeling approach depends on what is to be described and revealed [28]. Our goal in developing this framework was twofold: (1) to describe the behavioral aspect of the entire system—the human, the machine, and the environment, and (2) to use the description in order to reveal the potential for human-automation problems. One possible method for detecting such problems is by searching for mismatches between the states of the machine and the corresponding states of the human and environment [15]. It is our working hypothesis that most mode confusion problems occur (and therefore can be detected) at the point when the necessary synchronization between the modules is lost (c.f., [27]).

ANALYSIS OF A MODE-CONFUSION INCIDENT

During the field experiment described earlier, some 28 incidents involving mode confusion were observed and documented [8]. Using the Ofan framework we describe here one of these incidents. This particular one involved confusion regarding the speed of the aircraft (a Boeing B-757) while the crew was transitioning from one vertical mode to another.

The aircraft was climbing to 11,000 feet per air traffic control (ATC) instructions, and a fully automatic vertical mode called "Vertical Navigation" was active. In this mode the speed target value is obtained from the flight management computer, which computes the most economical speed (about 300 knots in this case). During the climb (at about 10,500 feet), ATC instructed the crew to reduce speed to 240 knots. The first officer engaged a unique feature of "Vertical Navigation" called "speed intervene," which allowed him to enter the speed, specified by ATC, via the mode-control panel as a new reference parameter (see Figure 3).

As the aircraft neared 11,000 feet, it started the level-off maneuver by automatically disengaging "Vertical Navigation" and engaging "Altitude Capture" mode. Once at 11,000 feet, the "Altitude Capture" mode disengaged and the "Altitude Hold" mode was engaged automatically. During and after the maneuver, the speed was kept at 240 knots. Shortly after, ATC instructed the crew to climb to 14,000. The first officer reached up to the mode-control panel and engaged the "Vertical Navigation" mode in order to initiate the climb. However, instead of climbing at a speed of 240 knots (as was still required by ATC), the aircraft speed defaulted to 300 knots! The crew violated the ATC speed restriction because they assumed that "Vertical Navigation" mode would "remember" the speed reference parameter entered previously into the mode-control panel. However, the "Vertical Navigation" mode, once re-engaged, defaulted to economy speed.

Description

The automatic flight control system of this aircraft has some eight modes to control the vertical aspect of the flight (see Figure 1). Three modes are discussed here: "Altitude Capture," "Altitude Hold"—ALT HOLD, and "Vertical Navigation"—VNAV (see Figure 3). With respect to the speed reference parameter (a constraint on mode behavior), the "Altitude Capture" and "Altitude Hold" modes obtain this parameter from the mode-control-panel. By default, the "Vertical Navigation" speed reference-parameter is obtained from the flight management computer (which computes economy speed). Yet another option, called "speed intervene," allows the pilot to override the flight management computer speed and enter a different speed. This is achieved by pressing the "speed knob" and then dialing in the desired speed into the mode-control panel.

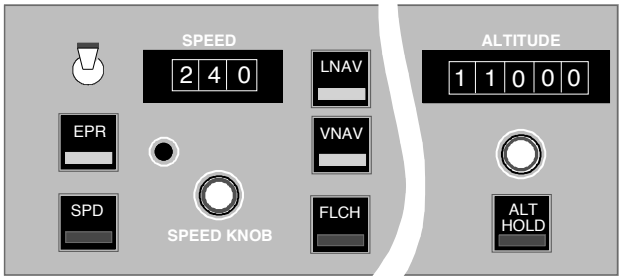

Figure 3. Mode-control panel

Model

Figure 4 depicts part of the Ofan representation of this example. The *Environment* module describes the demands (in this case ATC instructions) as a vector of several elements. For example, speed restrictions can be either instructed directly by the air traffic controller or mandated by a given procedure (e.g., a standard instrument departure). The *Human-Function/Tasks* module encompasses the two functions performed by the crew: (1) "modify," and (2) "monitor." The "modify" function describes two tasks (modify speed, change altitude) as states. The "monitor" function describes the two monitoring tasks as orthogonal states.

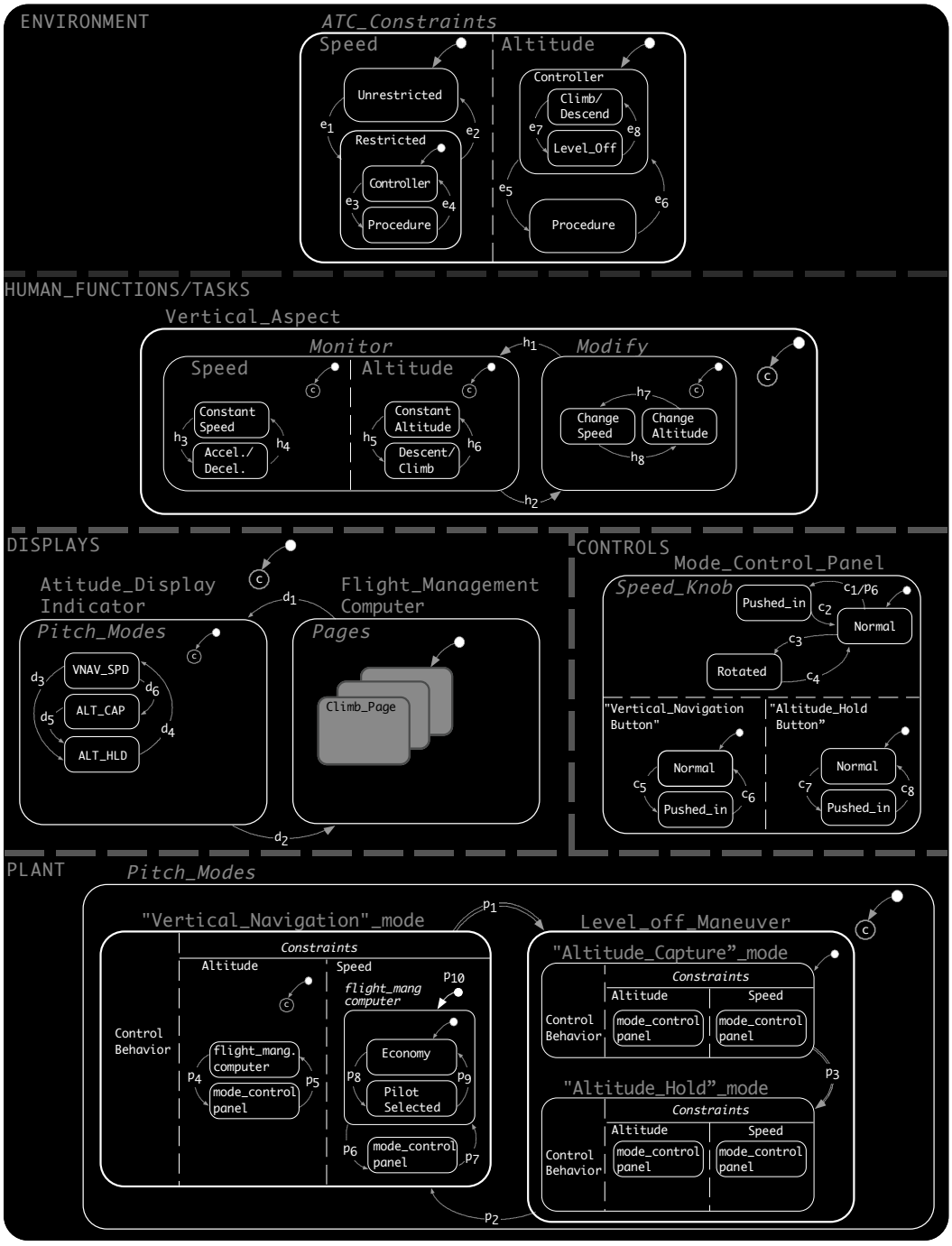

Figure 4. *"Ofan" representation.*

The *Controls* module depicts the mode-control panel and its various knobs and buttons: (1) the speed knob which, by pushing it, engages the "speed intervene" feature, and (2) buttons for manually engaging modes (see Figure 3). The *Plant* module represents the three modes discussed in this example: "Altitude Capture," "Altitude Hold," and "Vertical Navigation." Each mode is represented by two orthogonal sub-states: one indicating the control behavior of the mode, and the other listing its reference parameters (speed and altitude). For example, when in "Vertical Navigation" mode, the speed

reference parameter is either obtained from the flight management computer (default entry—depicted as a small arrow) or from the mode-control panel. When the source of reference parameter is the flight management computer, the speed can be either computed (economy speed) or selected (entered manually into the computer) by the pilot. Transitions, either automatic or manual, in the Plant module are between (1) "Vertical Navigation" mode and the level-off modes—an automatic transition (depicted as two parallel lines), (2) "Altitude Capture" mode and the "Altitude Hold" mode—also an automatic transition, and (3) "Altitude Hold" and "Vertical Navigation" mode—a manual transition. Lastly, the Displays module describes the annunciation of the mode status on the attitude-display indicator.

ANALYSIS

The mode problem occurred when air traffic control instructed the crew to climb from 11,000 to 14,000 feet. At that point, the crew re-engaged the "Vertical Navigation" mode. However, they forgot to *re*-enter the speed reference-parameter (240 knots). As a result, a speed violation occurred.

Once the automated flight control system transitions from any mode to "Vertical Navigation" mode, the speed target value defaults to the economy speed. This is represented in the "Vertical Navigation" mode description (Plant module) as the default transition into flight management system (p10). The result of this discrepancy—between the expected behavior (same speed) versus the actual behavior (speed change)—was a violation on part of the crew.

The problem is detected when we note a mismatch between the states of the *Environment* and *Human-Function/Tasks* modules and the state of the *Plant* module. While the *Environment and Human-Function/Tasks* modules did not change their corresponding states (speed is still restricted by air traffic control, and the crew is only required to monitor it), the *Plant* did. The system, as a whole, lost the necessary synchronization between its modules, and a mode confusion was born.

This specific mode problem, we argue, occurs because of a non-fluent mode transition [26]. While transitions among several modes all maintain a global reference parameter (speed), one transition does not. After transitioning into the "Vertical Navigation" mode, a default entry changes the value of the global reference parameter into a local one (economy speed). Furthermore, the layout of the controls and displays do not allude to this default transition.

We believe that this unique structural feature of the design, which lies latent within the software, may produce mode confusion when pilots are unaware of it. Furthermore, we hypothesize that this latent structural feature may produce mode confusion in any type of human-machine system. Therefore, the immediate goal of our research is to classify this and other type of structural features into a taxonomy of mode discrepancy. Using such a taxonomy, similar structural features found in a specification document of a new design can be identified, analyzed, and corrected.

CONCLUSIONS

> *Imagine an island with two mountains on it. A quantitative change, or rise, in the level of the ocean may convert this single island into two islands. This will happen at a point where the level of the ocean rises higher than the saddle between the two mountains. The qualitative pattern was latent before the quantity had impact on it...* (Gregory Bateson, *Mind and Nature*, 1979, p.58)

Our general approach for studying mode problems is to map human-automation interaction. With the aid of such a map, or a qualitative pattern in Bateson's terms, we can view the entire territory—the human functions and tasks, the plant's states, and the environment. Once we have a complete map we can also predict the behavior of the human-machine-environment system, given external and internal events. It may also allow us to identify some of the structural features of the design that produce unwanted results.

In this paper we described human interaction with complex system that employ modes using two reciprocal views: (1) summary analysis of mode occupancy and transitions, and (2) a formal specification of this interaction. The summary analysis provided a view, from outside the system, on how it is used. We describe pilots' transitions among the possible mode configurations of the automated flight control system of the Boeing B-757/767 aircraft. The pattern that emerged was constrained by the phases of flight, standard operating procedures, the demands from the operational environment (mainly ATC), and the mode configuration space of the system.

The formal specification of the human-automation interaction provides a view, from inside the system, of its internal mechanisms. In developing this specification framework, we have tried to captured the constraints on the system behavior and synthesize them into the framework. We describe the standard operating procedures and phase of flight in the *Human-Function/Tasks* module. The demands from ATC and the environment are captured in the *Environment* module. Finally, the mode configuration space and the interface with it are modeled in the *Plant* and *Control/Display* modules, respectively.

Our ongoing research effort is to develop a methodology for evaluating human-machine systems. The first step in this direction is presented here—a framework for modeling the human-machine environment system. The second step is to develop a taxonomy of structural features that induce mode confusion. The interim goal is to provide heuristic methods for early detection in a given design specification. Our long-term goal is to develop a general theory of mode problems that will go beyond a catalog of past causes.

Human interaction with complex and automated control systems is a non-trivial problem. It appears that its contribution to mishaps in domains that employ this technology is quite significant. In this paper, we briefly cited some symptoms of this problem. We then presented a two-way methodology for describing human-automation interaction. We argue that only a cross-disciplinary effort, integrating methods from control theory, system engineering, and human factors, will ever come close to ameliorating these chronic problems in human interaction with complex control systems.

ACKNOWLEDGMENTS

This work was supported by NASA's Aviation Safety and Automation Program. The first author was supported by grant NCC2-327 from NASA Ames Research Center to the San Jose State University Foundation. Part of this research was conducted at the Center for Human Machine System Research (CHMSR) Georgia Institute of Technology, Atlanta. The authors thank Michael Feary, James Lockhart, Christine Mitchell, Everett Palmer, and Alan Price for their valuable help in this effort. Special acknowledgment is due to 60 anonymous airline pilots who allowed us to collect data about their interactions with the automated flight control system.

REFERENCES

1. Ashby, R. W. (1956). *An introduction to cybernetics*. New York: John Wiley & Sons.
2. *Aviation Week and Space Technology*. (1994). Autopilot go-around key to China Air Lines crash. May 9, 31-32.
3. *Aviation Week and Space Technology*. (1995). Automated cockpits: who's in charge? Part 1 and 2. 142(5 and 6),
4. Commission of inquiry. (1990). *Air France Airbus A-320 F-GFKC, Mulhouse-Habsheim airport, June 26, 1988* (edited excerpts reprinted in *Aviation Week and Space Technology*, June 4, 1990). French Ministry of Planning, Housing, Transport and Maritime Affairs.
5. Commission of inquiry. (1992). *Air Inter Airbus A-320, F-GGED, Strasbourg-Entzheim airport, January 20, 1992* (edited excerpts reprinted in *Aviation Week and Space Technology*, March 13, 1995—vol. 142, number 9). French Ministry of Planning, Housing, Transport and Maritime Affairs.
6. Commission of inquiry. (1995). *Airbus Industrie A-330, production #42, Toulouse-Blagnac airport, France, June 30, 1994* (edited excerpts reprinted in *Aviation Week and Space Technology*, May 22, 1995—vol. 142, number 21). French Ministry of Planning, Housing, Transport and Maritime Affairs.
7. Degani, A., and Kirlik, A. (1995). Modes in Human-Automation Interaction: Initial Observations about a Modeling Approach. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics* (SMC), (pp. 3443-3450). Vancouver, Canada: IEEE.
8. Degani, A., Mitchell, C. M., and Chappell, A. R. (1995). Task models to guide analysis: Use of the operator function model to represent mode transitions. In R. S. Jensen (Ed.), *Proceedings of the Eighth International Aviation Psychology Symposium Conference*. Columbus, OH: The Ohio State University.
9. Degani, A., and Wiener, E. L. (1994). *On the design of flight-deck procedures* (NASA Contractor Report 177642). Moffett Field, CA: NASA Ames Research Center.
10. Eldredge, D., Dodd, R. D., and Mangold, S. J. (1991). *A review and discussion of flight management system incidents reported to the aviation safety reporting system.* Cambridge, MA: Volpe National Transportation Systems Center.

11. Gopal, B. S., and Rao, C. R. S. (1991). *Indian Airlines A-320 VT.EP, Bangalore, India, February, 14, 1990* (Report of the technical assessors to the court of inquiry). Indian Government.
12. Harel, D. (1987). Statecharts: A visual formalism for complex systems. *Science of Computer Programming, 8*, 231-274.
13. Harel, D. (1988). On visual formalisms. *Communications of the ACM, 31(5)*, 514-530.
14. Jones, P. M., Chu, R. W., and Mitchell, C. M. (1995). A methodology for human-machine systems research: Knowledge engineering, modeling, and simulation. *IEEE Transactions on Systems Man and Cybernetics, SMC-25(7)*, 1025-1038.
15. Kirlik, A. (1995). Requirements for psychological models to support design: Toward ecological task analysis. In J. M. Flach, P. A. Hancock, J. K. Caird and K. J. Vicente (Ed.), *An ecological approach to human machine systems I: A global perspective*. Hillsdale, NJ: Lawrence Erlbaum.
16. Lambergts, A. A. (1983). *Operational aspects of integrated vertical flight path and speed control system* (Society of Automotive Engineers [SAE] technical paper 831420). Warrendale, PA: Society of Automotive Engineers.
17. Mangold, S. J., and Eldredge, D. (1995). Flight management system information requirements. In R. S. Jensen (Ed.), *Proceedings of the Eighth International Aviation Psychology Symposium Conference*. Columbus, OH: The Ohio State University.
18. Mellor, P. (1994). CAD: Computer aided disasters. *High Integrity Systems, 1(2)*, 101-156.
19. Miller, R. A. (1979). *Identification of finite state models of human operations* (Dept. of Industrial and systems Engineering project # 784556 AFOSR #77-3152). Columbus, OH: The Ohio State University.
20. Minsky, M. L. (1967). *Computation: Finite and infinite machines*. Englewood Cliffs, NJ: Prentice-Hall.
21. Monk, A. (1986). Mode errors: a user-centered analysis and some preventative measures using keying-contingent sound. *International Journal of Man-Machine Studies, 24*, 313-327.
22. Moray, N., Lootsteen, P., and Pajak, J. (1986). Acquisition of process control skills. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-16(4)*, 497-504.
23. Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard university Press.
24. Norman, D. A. (1981). Categorization of action slips. *Psychological Review, 1(88)*, 1-15.
25. Norman, D. A. (1983). Design rules based on analysis of human error. *Communications of the ACM, 26(4)*, 254-258.
26. Palmer, E. A. (1995). "Oops, it didn't arm"—A case study of two automation surprises. In R. S. Jensen (Ed.), *Proceedings of the Eighth International Aviation Psychology Symposium Conference*. Columbus, OH: The Ohio State University.
27. Rasmussen, J. (1986). *Information processing and human machine interaction: An approach to cognitive engineering*. New York: Elsevier Science Publishing, Inc.

28. Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. (1991). *Object oriented modeling and design*. Englewood Cliffs, NJ: Prentice-Hall.

29. Sarter, N. B., and Woods, D. D. (1995). How in the world did we ever get into that mode? Mode error and awareness in supervisory control. *Human Factors, 37(1)*, 5-20.

30. Sellen, A. J., Kurtenbach, G. P., and Buxton, W. A. (1992). The prevention of mode errors through sensory feedback. *Human-computer interaction, 7*, 141-164.

31. Tukey, J. W. (1977). *Exploratory data analysis*. Reading, MA: Addison-Wesley.